

1      **CLAIMS**

2      1. One or more computer readable media having stored thereon a  
3      program that, when executed by one or more processors, causes the one or more  
4      processors to perform acts including:

- 5            identifying a plurality of key instructions in a function;  
6            inserting into the function, for each of the plurality of key instructions, an  
7            extra instruction that modifies a register based at least in part on the corresponding  
8            key instruction;  
9            identifying a set of inputs to the function; and  
10          determining a checksum for the function based at least in part on mapping  
11          contents of the register to the set of inputs.

12  
13      2. One or more computer readable media as recited in claim 1, wherein  
14      the identifying a plurality of key instructions comprises identifying, as a key  
15      instruction, each instruction in the function that possibly modifies a register or a  
16      flag.

17  
18      3. One or more computer readable media as recited in claim 1, wherein  
19      the identifying a plurality of key instructions comprises identifying, as the  
20      plurality of key instructions, a plurality of instructions that each modify one or  
21      more registers or one or more flags.

1       **4.** One or more computer readable media as recited in claim 1, wherein  
2 the inserting comprises inserting each extra instruction in a location within the  
3 function so that the extra instruction is executed if the corresponding key  
4 instruction is executed.

5  
6       **5.** One or more computer readable media as recited in claim 1, wherein  
7 the inserting comprises inserting each extra instruction in a location within the  
8 function so that the extra instruction is executed after the corresponding key  
9 instruction is executed.

10  
11      **6.** One or more computer readable media as recited in claim 1, wherein  
12 the inserting comprises inserting the extra instructions into the function without  
13 altering the control flow of the function.

14  
15      **7.** One or more computer readable media as recited in claim 1, wherein  
16 the inserting comprises inserting, for at least one of the plurality of key  
17 instructions, a plurality of extra instructions that modify one or more registers.

18  
19      **8.** One or more computer readable media as recited in claim 1, wherein  
20 the identifying a set of inputs comprises identifying a set of input patterns to the  
21 function that result in different valid computation paths in the function being  
22 taken.

1           9. One or more computer readable media as recited in claim 1, wherein  
2 the identifying a set of inputs comprises identifying a set of input patterns to the  
3 function that result in all valid computation paths in the function being taken.

4

5           10. One or more computer readable media as recited in claim 1, wherein  
6 the determining comprises determining as the checksum both an initial value ( $x_0$ )  
7 and a calculated value ( $Cks$ ), wherein the initial value is a first input of the set of  
8 inputs, and wherein the calculated value is calculated according to the following  
9 process:

10

```
Start with x = x0
Cks := f(x0) XOR x0
For i=1 to K do
    xi := g(f(xi-1))
    Cks += f(xi) XOR xi
End for
```

11           wherein  $K$  is the number of inputs in the set of inputs and  $g$  represents the mapping  
12 function.

13           11. One or more computer readable media as recited in claim 1, wherein  
14 the function is part of a software program.

15

16

17           12. A method comprising:

18           generating a checksum on bytes of a digital good without reading the bytes.

19

20

21

22

23           13. A method as recited in claim 12, wherein the generating comprises:  
24           identifying a plurality of key instructions in a function;

1                   inserting into the function, for each of the plurality of key instructions, an  
2 extra instruction that modifies a register based at least in part on the corresponding  
3 key instruction;

4                   identifying a set of inputs to the function; and  
5                   determining a checksum for the function by mapping contents of the  
6 register to the set of inputs.

7

8                  **14.** A method as recited in claim 13, wherein the identifying comprises  
9 identifying, as a key instruction, each instruction in the function that possibly  
10 modifies a register or a flag.

11

12                 **15.** One or more computer-readable memories comprising computer-  
13 readable instructions that, when executed by a processor, direct a computer system  
14 to perform the method as recited in claim 13.

15

16                 **16.** A method comprising:

17                   inserting, into a segment of a digital good, a plurality of instructions that  
18 modify a register;

19                   identifying a set of inputs to the segment; and  
20                   determining a checksum value for the segment based at least in part on both  
21 the set of inputs and the register contents.

22

23                 **17.** A method as recited in claim 16, wherein the inserting comprises:  
24                   identifying a plurality of key instructions in the segment; and

1                   inserting into the segment, for each of the plurality of key instructions, an  
2 extra instruction that modifies a register based at least in part on the corresponding  
3 key instruction.

4  
5                   **18.** A method as recited in claim 17, wherein the identifying a  
6 comprises identifying, as a key instruction, each instruction in the segment that  
7 possibly modifies a register or a flag.

8  
9                   **19.** A method as recited in claim 17, wherein the identifying a plurality  
10 of key instructions comprises identifying, as the plurality of key instructions, a  
11 plurality of instructions that each modify one or more registers or one or more  
12 flags.

13  
14                  **20.** A method as recited in claim 16, wherein the inserting comprises  
15 inserting each of the plurality of instructions in a location within the segment so  
16 that the instruction is executed if a corresponding key instruction is executed.

17  
18                  **21.** A method as recited in claim 16, wherein the inserting comprises  
19 inserting each of the plurality of instructions in a location within the segment so  
20 that the instruction is executed after a corresponding key instruction is executed.

21  
22                  **22.** A method as recited in claim 16, wherein the inserting comprises  
23 inserting the extra instructions into the segment without altering a control flow of  
24 the segment.

1           **23.** A method as recited in claim 16, wherein the inserting comprises  
2 inserting a plurality of extra instructions that modify one or more registers.  
3

4           **24.** A method as recited in claim 16, wherein the identifying a set of  
5 inputs comprises identifying a set of input patterns to the segment that result in  
6 different valid computation paths in the segment being taken.  
7

8           **25.** A method as recited in claim 16, wherein the identifying a set of  
9 inputs comprises identifying a set of input patterns to the segment that result in all  
10 valid computation paths in the segment being taken.  
11

12           **26.** A method as recited in claim 16, wherein the determining comprises  
13 determining as the checksum value both an initial value ( $x_0$ ) and a calculated value  
14 ( $Cks$ ), wherein the initial value is a first input of the set of inputs, and wherein the  
15 calculated value is calculated according to the following process:  
16

```
Start with x = x0
Cks := f(x0) XOR x0
For i=1 to K do
    xi := g(f(xi-1))
    Cks += f(xi) XOR xi
End for
```

20           wherein  $K$  is the number of inputs in the set of inputs and  $g$  represents the mapping  
21 function.  
22  
23  
24  
25

1           **27.** A method as recited in claim 16, wherein the digital good comprises  
2 a software program.  
3

4           **28.** One or more computer-readable memories comprising computer-  
5 readable instructions that, when executed by a processor, direct a computer system  
6 to perform the method as recited in claim 16.  
7

8           **29.** A production system, comprising:  
9           a memory to store an original program; and  
10          a production server equipped with an oblivious checking protection tool  
11         that is used to augment the original program for protection purposes, the  
12         production server being configured to identify a plurality of segments in the  
13         original program and apply oblivious checking to each of the plurality of  
14         segments.  
15

16           **30.** A production system as recited in claim 29, wherein the production  
17         server is to apply oblivious checking to each of the plurality of segments by:  
18           inserting, into the segment, a plurality of instructions that modify a register;  
19           identifying a set of inputs to the segment; and  
20           determining a checksum value for the segment based at least in part on both  
21         the set of inputs and the register content that results from applying the set of inputs  
22         to the segment.  
23

1           **31.** A production system as recited in claim 29, wherein the inserting  
2 comprises:

3                 identifying a plurality of key instructions in the segment; and  
4                 inserting into the segment, for each of the plurality of key instructions, an  
5 extra instruction that modifies a register based at least in part on the corresponding  
6 key instruction.

7

8           **32.** A production system as recited in claim 31, wherein the identifying  
9 a comprises identifying, as a key instruction, each instruction in the segment that  
10 possibly modifies a register or a flag.

11

12           **33.** A production system as recited in claim 29, wherein the inserting  
13 comprises inserting the extra instructions into the segment without altering a  
14 control flow of the segment.

15

16           **34.** A client-server system, comprising:  
17                 a production server to apply oblivious checking to a program to produce a  
18 protected program; and  
19                 a client to store and execute the protected program, the client being  
20 configured to evaluate the protected program to determine whether the protected  
21 program has been tampered with.

1       **35.** A client-server system as recited in claim 34, wherein the  
2 production server is to apply oblivious checking to the program by:

3              inserting, into a segment of the program, a plurality of instructions that  
4 modify a register;

5              identifying a set of inputs to the segment; and

6              determining a checksum value for the segment based at least in part on both  
7 the set of inputs and the register content that results from applying the set of inputs  
8 to the segment.

9  
10       **36.** A client-server system as recited in claim 34, wherein the client is to  
11 evaluate the protected program by:

12              generating a checksum value for each of a plurality of segments of the  
13 program based at least in part on both a set of inputs to the segment and the  
14 content of a register that results from applying the set of inputs to the segment;

15              comparing, for each of the plurality of segments, the generated checksum  
16 value to a stored checksum value corresponding to the segment; and

17              determining the protected program has been tampered with if the generated  
18 checksum value for any one or more of the plurality of segments does not match  
19 the stored checksum value for the segment.

20  
21       **37.** A client-server system as recited in claim 34, wherein the program  
22 comprises a software program.

1           **38.** One or more computer readable media having stored thereon a  
2         plurality of instructions that, when executed by one or more processors, causes the  
3         one or more processors to perform acts including:

4             generating a checksum value for a segment of a digital good based at least  
5         in part on both a set of inputs to the segment and the content of a register that  
6         results from applying the set of inputs to the segment;

7             comparing the generated checksum value to a stored checksum value  
8         corresponding to the segment; and

9             determining that the digital good has been tampered with if the generated  
10       checksum value does not match the stored checksum value.

11  
12           **39.** One or more computer readable media as recited in claim 38,  
13         wherein the plurality of instructions further cause the one or more processors to  
14         perform acts including repeating the generating, comparing, and determining for a  
15         plurality of segments of the digital good.

16  
17           **40.** One or more computer readable media as recited in claim 38,  
18         wherein the digital good comprises a software program.